Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 52 (2016) 222 – 227

www.elsevier.com/locate/procedia

Changeable, Agile, Reconfigurable & Virtual Production

Development of a Web Based Monitoring System for a Distributed and Modern Production

Michael Stangl, M.Sc.[a], Julia Pielmeier, M.Sc.[a], Christoph Berger, M.Sc.[a], Dr.-Ing. Stefan Braunreuther[a], Prof. Dr.-Ing. Gunther Reinhart[b]

[a]*Fraunhofer IWU, Projektgruppe RMV, Beim Glaspalast 5, 86153 Augsburg, Germany*
[b]*Institut für Werkzeugmaschinen und Betriebswissenschaften Technische Universität München, Boltzmannstr.15 85748 Garching*

* Corresponding author. Tel.: +49-821-56883-123; fax: +49-821-56883-50. E-mail address: christoph.berger@iwu.fraunhofer.de

Abstract

Web technologies have experienced a rapid development in recent years. In particular web browsers enhanced their abilities because of the improvement of JavaScript, CSS3 and HTML5. Hence, richer web-based software solutions with an increasing range of functions are available. By using responsive web design (RWD), a technology to display content without resizing on different screens, developers are able to support a diverse range of devices with small effort.
In order to enable a monitoring of the current status of a production system, signals of many different sensors, machine and production data are required. Combining microcontrollers with sensors to embedded sensors enables an efficient way to communicate with web services. Due to the strong decline of prices for semiconductor technologies, companies are able to set up production machines with these technologies at low costs.
This paper presents a way to set up a distributed manufacturing control system by using common web technologies like RWD and embedded systems. We discuss advantages and drawbacks of web-based software solutions and show a methodical approach for the use in a modern production system. Finally, the functionality of the method is proven within an application example.

Peer-review under responsibility of the scientific committee of the Changeable, Agile, Reconfigurable & Virtual Production Conference 2016

Keywords: Manufacturing, Open architecture, Monitoring, Cyper Physical Systems

1. Introduction

The overall objective of this paper is to design a platform using common internet technologies in the area of cyber - physical production systems (CPPS). The described technical and architectural requirements have to be provided in order to realize:

- real time production planning and control by using sensor and event based algorithms,
- risk analysis and
- predictive maintenance for an integrated quality assurance.

These three tasks are offered as services and allow independent and flexible responses to changing environmental conditions. Therefore, a powerful software and hardware architecture is presented, which provides smart services on the basis of sensor data and a robust real-time communication. Starting from this communication architecture, the various services, such as event-based production planning and control, are conceptualized and the matching protocols for the communication are selected. The real-time requirements of industrial use cases are a crucial point, when using cloud services. For this reason, a short definition and an introduction to the limitations of real-time are given.

In total this paper presents an approach for a distributed manufacturing control system by using common web

Peer-review under responsibility of the scientific committee of the Changeable, Agile, Reconfigurable & Virtual Production Conference 2016
doi:10.1016/j.procir.2016.07.073

technologies like RWD and embedded systems. Finally, an exemplary implementation shows the feasibility of the concept.

1.1. Monitoring

Monitoring, by definition, means: systematic observing, detecting or measuring operations of a system and to detect changes [1]. It covers the detection of changes over a certain period of time using technical tools. It is essential, that data is recorded repeatedly at regular intervals. Thus, comparable conclusions can be obtained from the data.

Monitoring implies not only to observe a process. If there are any deviations from the desired course, it can be intervened in order to achieve a regular operation.

1.2. Soft- and Hardware Architecture

It is necessary to develop a suitable software and hardware architecture for Cyber-Physical Production Systems. Special challenges for the design of the architecture emerge because of the horizontal and vertical networking. The vertical networking allows single products to communicate with centrally provided services. This service can be made available within a company or across companies along the value chain [2].

Different architectures have already been developed in the field of research like shown in [3] or [4]. However, reuse of these architectures is not possible due to a lack of implementation templates. For a service-oriented platform the basic technologies such as the communication protocols (e. g. OPC UA, HTTP, …) have to be defined [5, 6]. Furthermore, safety aspects are becoming a focus for such designs [7]. On the one hand the functional safety architecture like the reliability of the system is important, on the other hand unauthorized access has to be prevented [7].

In summary it can be noticed, that the existing approaches of software and hardware architecture are not enough to meet the requirements of Cyber Physical Production Systems. Therefore, further research is required on architecture for a performant and robust information transmission of smart services.

1.3. Communication protocols

The resources and objects of a networked production have to communicate with each other and exchange data. The Transmission Control Protocol/Internet Protocol (TCP / IP) is a vendor independent communication protocol, which is a de facto standard for computer communication nowadays [8]. The reference model consists of four superposed layers [9].

The bottom layers 1 - 3 are responsible for the physical hardware connection, addressing and data transmission. The top level contains the Application layout. The most common protocol of this layer is the Hypertext Transfer Protocol (HTTP), which is the basis for the World Wide Web. Below the HTTP protocol and the increasingly used OPA - UA protocol are described.

1.3.1. HTTP

HTTP is a connection-oriented and stateless request-response protocol, where neither client nor server store information for further queries. A request is completely processed with a response [8]. Each HTTP message consists of two parts: the header, which contains information about the message and the body with the actual message.

In order to enable a tap-proofed communication, the HTTP Secure (HTTPS) protocol was developed. It is technically identical to HTTP, beside the difference that the communication takes place encrypted.

Both protocols distinguish between different methods. The most commonly used are described in Table 1.

Table 1. Common used HTTP methods

Method	Description
GET	By GET a resource is requested. Opening a web page, the browser performs the GET method.
POST	POST transfers data to the server and creates a new resource. E.g. sending contact form on a webpage sends an email.
PUT	With PUT the server changes an original resource. Can be utilized as “update” of content.

1.3.2. OPC-UA

The Open Platform Communication (OPC) Foundation created the same named OPC and standardized software interface which realizes the exchange of machine data between devices of different manufacturers. For some years exists the new OPC – Unified Architecture (OPC-UA) protocol which is standardized in IEC 62541. It is based on TCP and uses an optimized binary protocol in the application layer [10]. The web service is supported optional via HTTP (S), which facilitates the connection to browser applications.

Great emphasis has been put on the security aspects. Thus, on the one hand there is a mechanism, which denies unauthorized access to data and takes care that transmissions occur encrypted. On the other hand, it is ensured, that data is kept consistent. A manipulation or modification of process data is complicated by internal testing mechanisms.

1.4. Real time

To integrate intelligent objects/CPS in the industrial production environment, the requirements in the different production levels have to be fulfilled. Today, most production sites maintain roughly three levels, from the shop floor to the production (Manufacturing Execution Systems (MES)) up to the company management (Enterprise Resource Planning (ERP)) level. One reference architecture used to describe this structure is the information pyramid of automation [11]. The various tasks at each level lead to the different perception of the term “real time”.

Real time is defined by the ISO/IEC 2382 standard as the process of a computing system in which programs for processing accruing data are constantly ready. The term “ready” indicates that the processing results are available within a predetermined period [12]. A real-time system is

therefore a system which is able to process a specific task within a specific time window. In this case, it is not crucial to finish a task as soon as possible, it is more important to end at a fixed predefined time constraint. The execution of a task is fulfilled in time if the corresponding time requirement has been complied with. In addition, a distinction is made between hard and soft real-time requirements, seen in Figure 1.

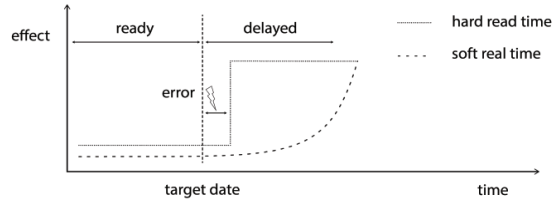

Figure 1 Illustration of the hard and soft real-time requirement [13]

In a periphery with a hard real-time condition (A) an unmet target date (d) leads to an unacceptable error, see Figure 1. With respect to the production, hard real-time conditions, such as turning, can be found on the shop floor level [13]. The sum of the start date (r) and the delta e (Δe) must always be less than the target date (d) in order to meet the hard real-time requirements, as shown in Formula (1).

$$A \equiv r + \Delta e \leq d \quad (1)$$

IEC 61784-2 (DIN EN 61784-3) proposes three real-time classifications, especially for the shop floor level. It refers to the maximum response time and also explicitly to suitable automation functions. For each real-time classification, Formula (2) has to be fulfilled. The hard real-time condition is characterized by the fact that the time condition A necessary at favorable conditions B, i.e. with probability (P) 1, is to be complied with.

$$P(A \mid B) \leq 1 \quad (2)$$

Soft real-time conditions allow the exceedance of a time limit as the target date (d). The system tolerates this condition and will continue to function, see Figure 1. The longer the time limit is exceeded, the greater the negative impact on the system. In production, we find some soft real-time conditions. One example is when a user starts a process and there is a delay between the user start time and the process start time.

In addition to the described timeliness, a real-time system must also have the property of predictability and determinism. Therefore, with regard to all information handling, we need a definition of the real-time requirement. That includes the processing time of the task until it receives the forwarded information from other systems.

1.5. Responsive web design (RWD)

RWD is a technology, to make websites flexible and reactive to adapt to the available display size [14]. It uses media queries with fluid, proportion-based grid systems and flexible images [15]. This allows the support of various screens, from small smartphones to large desktop monitors.

2. Approach

2.1. Network Architecture

A production monitoring systems requires current data of the production. For a web based monitoring of a distributed production, the required data must be sent via Internet. Therefore, an architecture is needed, which enables a secure, fast and reliable exchange of data.

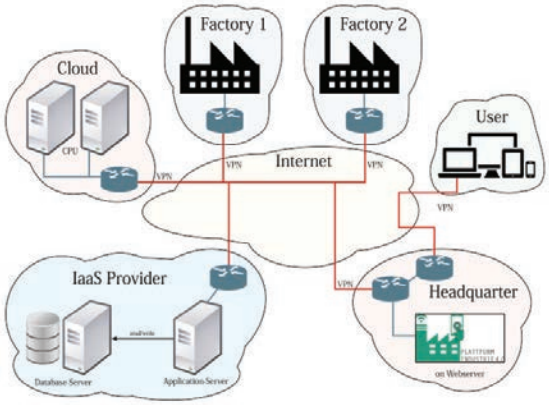

Figure 2 Network architecture

Figure 2 shows possible hybrid cloud solution for a networked production. It relies on cloud- and company network technologies. Based on a study of McKinsey, transferring infrastructure to a cloud can yield cost savings of over 50% [16]. Cloud services are easy to scale and adapt to current business requirements [17]. Regarding financial reasons and for maintaining flexibility most of the compute power, databases, storage and the related network components are rented from an Infrastructure as a Service (IaaS) provider.

Communication takes place via Internet. To enable a secure transfer of data between locations, Virtual Private Network (VPN) tunnels (see red lines) are used. It's a common solution to connect participants around the world to an extended private network by a secure link. Each site has routers to establish a connection to the company VPN. The VPN tunnel is used to send data from sensors to an Application Server (AS). Additionally, the transfer is SSL secured to gain a higher level of security. At first, each sensor has to authenticate to the AS. Afterwards the data is subjected to a plausibility check. Last step is saving the values in a database.

The available data is used to calculate indicators, predictive maintenance etc. These calculations can be performed by large data centers in a short time. Services like AWS-Lambda or Google Cloud Functions can use provided algorithms to process the given data. Billing is based on effectively required computing time which keeps the costs low.

The monitoring system is a web based browser application in the form of a Management Cockpit (MC) and has to be installed on a web server. It comes with a comprehensive role and rights management allowing a reliable user specific activation of functions. The Communication with the AS takes place via a SSL encrypted HTTPS connection. It's possible to attach a global accessible IP and domain, but for more security, access is only possible from inside the company's VPN.

IaaS provider typically charge fees for each VPN connection. To prevent high costs, the headquarter has a second router. It is used to offer employees being in remote networks a free VPN connection to the corporate network. The further communication takes place via the existing VPN tunnel.

2.2. Real time Monitoring Problems

As described in 1.4, the defined period of time to be real time depends on the considered application. Regarding a quality score, the value can be a few minutes old and still fulfil the requirements of real time. Though with respect to current production data, for example an injection molding process, which is completed in a few seconds, the required real time reduces significantly.

Since the MC is installed at a remote location, the data has to be transmitted over a long distance. Therefore, purely physical a certain time is needed. Since several years, the global transmission of data via Internet is carried out with fiber optic cables and laser light. The speed of light is approximately 300.000 km/h. In a fiber optic the speed is diminished to about 200.000 km/h [18].

Having a production facility in Sydney, but the AS in Frankfurt at a rented data center, the linear distance would be approx. 16.500 km. With Formula 3 the single signal propagation delay between the two sites can be calculated.

$$t = \frac{16.500\,km}{200.000\,km/s} = 82.5\,ms \qquad (3)$$

In practice, the time needed to transmit data is much higher, because several internet exchange points (IXP) have to be passed. Every IXP needs time for determining the next route point and for forwarding the data. Furthermore, connections are not direct and sometimes data is routed through the USA. There is a lot of capacity available, because of the very good infrastructure they build over the last years. The physical distance can be more than doubled because of this.

Additional time is added to the signal propagation delay because the AS needs time to process and save the incoming data. Also information has to be sent from AS to the MC user, who is placed at a different location.

To measure real signal propagation times, a so-called Ping-test can be performed. A client sends small data packages (32 bytes) to a server, who automatically responds. The passed time (for both ways) is measured. It should be noted, that real data is usually larger, which means it'll need more time for transmission. With a ping, the minimum necessary time can be measured reliably. This data helps to draw conclusions about the quality of the transmission and to estimate if real-time applications are possible.

Table 2 shows runtime measurements from different locations around the world to Frankfurt. A self-test from another location in Frankfurt was also executed. To obtain meaningful data, the test must be repeated several times on different days and hours.

Table 2. Ping test from different Locations to Frankfurt

Location	Min. [ms]	Max. [ms]	Avg. [ms]
Frankfurt – DE	1	2	1
Munich – DE	11	11	11
Toronto – CA	92	93	93
Los Angeles – US	150	151	150
Tokyo – JP	240	240	240
Sydney – AUS	323	324	323

Assuming that the AS is in Frankfurt, the headquarter in Munich and a production facility in Sydney, the earliest signal in Munich will be received 334 ms after it has been sent in Sydney (323ms Sydney → Frankfurt + 11 ms Frankfurt → Munich). This time doesn't contain the period the AS needs to process the incoming information. Besides, if the operator is not located in the headquarter, additional time is added.

If it's not possible to comply with a real-time condition, because of the long signal propagation delay, the location of the AS and of the MC can be moved. IaaS provider usually operate multiple data centers on every continent. This allows an easy shifting of the structure to get shorter latencies. Figure 3 shows a section of a modified Architecture. The production is located in the United States and Asia, while the headquarter is in Europe.

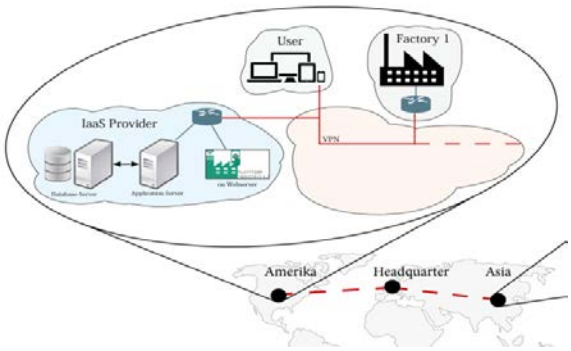

Figure 3 Worldwide network architecture

Both AS and database are replicated and are installed in the respective nearest data center. The MC is also outsourced to the cloud. This leads to very short distances (MC and AS even in the same data center) and thus to short signal propagation times. Access from the headquarter is still possible through the existing VPN tunnel, but there is no guarantee to get real

time data. The operator should be within the same region, to receive data in real time.

2.3. Data Transfer

The Application Server provides a RESTful web service. The communication follows the request/response principle of the HTTP protocol (see 1.3.1). This is due to the VPN tunnel, the data transfer is already secured against eavesdropping from external persons. However, participants in the corporate network could listen to it. To prevent this leak of information, the encrypted HTTPS protocol is entirely used for transferring data.

The RESTful interface mostly works with the HTTP methods, mentioned in chapter 1.3.1. The MC mainly uses GET when requesting information about the production. The sensors use the POST method, to send their data to the server. When an operator sends commands to machines or shifts the production sequence, the PUT method is used. By detecting the used method, the AS can distinguish the selected action and perform with the right behavior.

The server interface accepts and sends data formatted in XML or JSON text. Both formats are very popular with web developers [19] and often used, which ensures a long term support. However, the JSON Format is preferred. The size of the data to be transmitted is reduced considerably, because no closing tags must be used by declaring values [20]. Transferring a smaller amount of data is not only faster, it also helps saving costs, because IaaS provider charge data transfer fees. Table 3 shows differences in size between JSON and XML formatted data. These are average values for real measured data transfer from our case study.

Table 3. Packet size differences of XML and JSON text

Case	XML [bytes]	JSON [bytes]	Diff. [bytes]	Diff. [%]
Indicator	107	50	57	-53.3
Sensor data	375	145	230	-61.3
Event list	3509	2427	1082	-30.1

Regarding absolute values, the differences in size are very low (a few hundred bytes for sensor data). However, it's very important to take a look at the number of requests which will be sent every day. To calculate reliable and accurate predictive maintenance values, the chronological course of sensors is very important. Furthermore, a variety of sensors (sound, temperature, humidity, etc.) is needed [21]. The sensors send their data in very short ranges to the AS. Even for a very small production, more than 2 million requests can be sent during an average work day. By using JSON notation instead of XML, it is possible to save up to 500 MB of data per day.

Googles Cloud Platform currently charges 0.12 € / GB of transferred data [22]. Regarding the mentioned example, this would lead to an increase of cost of about 6 cents / day, when submitting XML formatted sensor data. Since JSON notation has no disadvantages with respect to quality of data, this form of transmission is preferably used. However, the XML format is also supported, to give more compatibility to other systems.

Modern machines have often integrated sensors and OPC-UA servers which use this technology for data exchange. For direct support of modern devices, an OPC-UA service can be implemented to the AS. Another technology is shown by [23]. They created a RESTful extension for OPC-UA to enable communication with RESTful web services.

3. Case Study

The *Lernfabrik für vernetzte Produktion* (LVP) at Fraunhofer IGCV in Augsburg shows a modern production presented by the example of a gearbox. For monitoring the production, a browser based control system is used.

Figure 4 Management Cockpit

Figure 4 shows the landing page of the MC based on the approach for large desktop browsers. The navigation and settings menu are not visible. The current layout is divided in two parts. The upper part contains selected indicators. The modular development allows an easy adaption of the layout to a user request. The MC is multilingual, so every user can choose his preferred language. The main view contains the floor plan for the selected production facility. The colors, based on traffic lights, offer a quick overview for every workplace. The view can be switched between *Production Control*, *Condition Monitoring* and *Risk Management*. Detailed information about assembly times, inventories, engine data etc. is provided by clicking on the respective box.

The design was built with the help of Twitter Bootstrap. It is a responsive framework which reduces the effort to support a wide range of devices with different screen sizes. Due to the fact, that the shopfloor is very small, there is no zoom function implemented. On very small devices this can diminish the user experience. When having very large production sites, a zoom is mandatory to retain a good usability for all devices.

The network architecture of the platform is set up as shown in Figure 2. The IaaS data center is located in Munich, approx. 100 km away from Augsburg. A large IXP (DE-CIX Munich) is located in Munich, which helps getting a low latency for transferring data.

Many productions use machines without integrated sensor systems. But they can be equipped with single board computers like a Raspberry Pi and some sensors. This makes them capable to communicate with the modern production monitoring. The Pi's are run by a Linux minimal operating system without graphical user interface. The source code for reading data from the sensors is written in C. Processing and submitting the information to the AS is executed by a Python script. Reaction tests show, that the average time to detect and process a sensor value is about 160 µs. To shorten the reaction time, a complete implementation in C would be necessary. The average time for sending data to the AS is 132ms. Regarding this production, practice has shown that a delay of approx. 200ms is sufficient, to not risk the extension of cycle times.

The AS has the possibility to push data to the MC through the Web Sockets protocol. This is a technology to keep persistent connections between clients and server. Through this channel, the server has the possibility to send messages to a connected client [24]. When detecting new events, the AS can send information about production, critical situations etc. to the clients. This leads to a strong decrease of requests, which reduces network traffic and server load.

The user has the possibility to activate several messaging channels. It is possible to receive push notifications. However, when being offline, no message can be sent. If critical errors appear, the AS can turn on an alarm signal or send an SMS to an operator.

4. Summary

In this paper we presented a way to develop a web based monitoring for a distributed production. By using cloud services effectively, costs can be sharply reduced. However, latency becomes very important regarding real time applications, when data centers are no longer in house.

The developed MC for LVP shows that a RWD design is useful for supporting a variety of devices. The development effort can be sharply reduced although platform independent usage is possible. The versatile architecture can be applied to a strong and reliable cloud infrastructure which makes real time monitoring possible. A future work should concern on integrating an OPC-UA service to increase the compatibility of the whole system.

Acknowledgements

The research and development project OpenServ4P is funded by the German Federal Ministry for Economic Affairs and Energy within the Framework Concept "Smart Service World" and managed by the Project Management Deutsches Zentrum für Luft- und Raumfahrt e. V. (DLR)

More information about the ongoing research project is provided via the website: www.openserv4p.de.

References

[1] Beims, M., 2012. *IT-Service Management mit ITIL®: ITIL® Edition 2011, ISO 20000:2011 und PRINCE2® in der Praxis*, 3rd edn. Hanser, München.

[2] VDI/VDE. Referenzarchitekturmodell Industrie 4.0 (RAMI4.0). https://www.vdi.de/fileadmin/user_upload/VDI-GMA_Statusreport_Referenzarchitekturmodell-Industrie40.pdf. Accessed 28 May 2016.

[3] Vogel-Heuser, B., Bender, K., Kegel, G., Wucherer, K., 2009. Global Information Architecture for Industrial Automation, p. 108.

[4] Reinhart, G., Engelhardt, P., Geiger, F., Philipp, T.R. *et al*., 2013. Cyber-Physische Produktionssysteme: Produktivitäts- und Flexibilitätssteigerung durch die Vernetzung intelligenter Systeme in der Fabrik, p. 84.

[5] Mahnke, W., Damm, M., Leitner, S.-H., 2009. *OPC Unified Architecture: How randomness acts in time*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg.

[6] Richardson, L., Ruby, S., 2007. *Web Services mit REST*, 1st edn. O'Reilly, Beijing.

[7] Broy, M., 2010. *Cyber-Physical Systems*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg.

[8] Mandl, P., Bakomenko, A., Weiß, J., 2010. *Grundkurs Datenkommunikation*, 2nd edn. Vieweg+Teubner Verlag / GWV Fachverlage GmbH Wiesbaden, Wiesbaden.

[9] Solomon, M.G., Kim, D., Carrell, J.L., 2015. *Fundamentals of communications and networking*. Jones & Bartlett Learning, Burlington MA.

[10] OPC Foundation. Wegbereiter der 4. industriellen (R)Evolution. https://opcfoundation.org/wp-content/uploads/2014/03/OPC_UA_I_4.0_Wegbereiter_DE_v2.pdf. Accessed 19 May 2016.

[11] Schlechtendahl, J., Keinert, M., Kretschmer, F., Lechler, A. *et al.* Making existing production systems Industry 4.0-ready 9, p. 143.

[12] ISO. Information technology - Vocabulary, Genf. International Organization for Standardization, 2015(DIN ISO/IEC 2382). http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=63598. Accessed 27 May 2016.

[13] Zöbel, D., 2008. *Echtzeitsysteme: Grundlagen der Planung*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg.

[14] Zillgens, C., 2013. *Responsive Webdesign*. Hanser, München.

[15] De Graeve, K. HTML5 - Responsive Web Design. https://msdn.microsoft.com/en-us/magazine/hh653584.aspx. Accessed 30 May 2016.

[16] Diamadi, Z., Dubey, A., Pleasance, D., Vora, A. Winning in the SMB Cloud: Chariting a path to success. https://www.mckinsey.com/…/Winning_in_the_SMB_Cloud.ashx. Accessed 24 May 2016.

[17] Mahmood, Z., 2013. *Cloud computing: Methods and practical approaches*. Springer, New York, NY.

[18] Oliviero, A., Woodward, B., 2009. *Cabling: The complete guide to copper and fiber-optic networking*, 4th edn. Wiley Pub, Indianapolis, IN.

[19] Abeysinghe, S., 2008. *RESTful PHP web services: Learn the basic architectural concepts and steps through examples of consuming and creating RESTful web services in PHP*. Packt Publishing Ltd, Birmington, U.K.

[20] Solutions, K., 2008. *Ajax Black Book, New Edition (With Cd)*. Dreamtech Press.

[21] Pouliezos, A.D., Stavrakakis, G.S., 1994. *Real time fault monitoring of industrial processes*. Kluwer Acad. Publ, Dordrecht.

[22] Google. Google Compute Engine Pricing. https://cloud.google.com/compute/pricing. Accessed 25 May 2016.

[23] Gruner, S., Pfrommer, J., Palm, F. A RESTful extension of OPC UA, in *2015 IEEE World Conference on Factory Communication Systems (WFCS)*, p. 1.

[24] Wang, V., Salim, F., Moskovits, P., 2013. *The definitive guide to HTML5 WebSocket*. Apress, New York, s.l.